A SECRET SHARING SCHEME BASED ON GROUP PRESENTATIONS AND THE WORD PROBLEM

MAGGIE HABEEB, DELARAM KAHROBAEI, AND VLADIMIR SHPILRAIN

ABSTRACT. A (t, n)-threshold secret sharing scheme is a method to distribute a secret among n participants in such a way that any t participants can recover the secret, but no $t-1$ participants can. In this paper, we propose two secret sharing schemes using non-abelian groups. One scheme is the special case where all the participants must get together to recover the secret. The other one is a (t, n)-threshold scheme that is a combination of Shamir's scheme and the group-theoretic scheme proposed in this paper.

1. Introduction

Suppose one would like to distribute a secret among n participants in such a way that any t of the participants can recover the secret, but no group of $t-1$ participants can. A method that allows one to do this is called a (t, n)-threshold scheme. Shamir [7] back in 1979 offered a (t, n)-threshold scheme that utilizes polynomial interpolation. In Shamir's scheme the secret is an element $x \in \mathbb{Z}_p$. In order to distribute the secret, the dealer begins by choosing a polynomial f of degree $t-1$ such that $f(0) = x$. Then he sends the value $x_i = f(i)$ secretly to participant P_i. In order for the participants to recover the secret, they use polynomial interpolation to recover f and hence the secret $f(0)$. In this setting, no $t-1$ participants can gain any information about the secret while any t of them can (see [7] or [8]).

The field of non-commutative group-based cryptography has produced many new cryptographic protocols over the last decade or so. We refer an interested reader to [4] or [1] for a survey of developments in this area. Recently, Panagopoulos [6] suggested a (t, n)-threshold scheme using group presentations and the word problem. His scheme is two-stage: at the first stage, long-term private information (defining relations of groups) is distributed to all participants over secure channels. At the second stage, shares of the actual secret are distributed to participants over open channels. Thus, an advantage of Panagopoulos' scheme over Shamir's scheme is that the actual secret need not be distributed over secure channels. This is obviously useful in various real-life scenarios. On the other hand, his scheme is not quite practical because in his scheme, it takes time exponential in n to distribute the shares of a secret to n participants.

In this paper, we present two practical secret sharing schemes based on group presentations and the word problem. The first one is a two-stage scheme, like the one due to

Research of Delaram Kahrobaei was partially supported by a PSC-CUNY grant from the CUNY research foundation, as well as the City Tech foundation.

Research of Vladimir Shpilrain was partially supported by the NSF grants DMS-0914778 and CNS-1117675.

Panagopoulos mentioned above, but it is just an (n, n)-threshold scheme. Our second scheme is a (t, n)-threshold scheme, $0 < t \leq n$. Both our schemes are designed for the scenario where the dealer and participants initially are able to communicate over secure channels, but afterwards they communicate over open channels.

We first consider the special (trivial) case where $t = n$; that is, we propose a scheme in which all participants are needed to recover the secret. Then, we propose a hybrid scheme that combines Shamir's scheme and the idea of the (n, n)-threshold scheme we proposed. This combined scheme has the same distributed secret as Shamir's scheme does, but rather than sending $f(i) = x_i$ over secure channels we send the integers in disguise, and then participants use group-theoretic methods to recover the integers. This scheme has the following useful advantages over Shamir's scheme:

- The actual secret need not be distributed over secure channels and, furthermore, once the long-term private information is distributed to all participants, several different secrets can be distributed without updating the long-term private information.
- While recovering the secret, participants do not have to reveal their shares to each other if they do not want to.

2. Very Brief Background on Group Theory

In this section, we give a minimum of information and notation from group theory necessary to understand our main Sections 3 and 4. Further facts from group theory that are used, explicitly or implicitly, in this paper are collected in Sections 6 and 7.

A free group $F = F_m$ of rank m, generated by $X = \{x_1, \ldots, x_m\}$, is the set of all reduced words in the alphabet $\{x_1^{\pm 1}, \ldots, x_m^{\pm 1}\}$, where a word is called reduced if it does not have subwords of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$.

Any m-generated group G is a factor group of F_m by some normal subgroup N. If there is a recursive (better yet, finite) set of elements $\{r_1, \ldots, r_k, \ldots, \}$ of N that generate N as a normal subgroup of F_m, then we use a compact description of G *by generators and defining relators*:

$$G = \langle x_1, \ldots, x_m \mid r_1, \ldots, r_k, \ldots, \rangle$$

and call $\{x_1, \ldots, x_m\}$ *generators* of G, and $\{r_1, \ldots, r_k, \ldots, \}$ *defining relators* of G. Here "generate N as a normal subgroup of F_m" means that every element u of N can be written as a (finite) product of conjugates of relators r_i: $u = \prod_k h_k^{-1} r_{i_k} h_k$.

Given a presentation of a group G as above, the *word problem* for this presentation is: given a word $w = w(x_1^{\pm 1}, \ldots, x_m^{\pm 1})$ in the generators of G, find out whether or not $w \in N$. (If $w \in N$, we say that $w = 1$ in G.)

3. An (n, n)-threshold Scheme

Suppose we would like the dealer to distribute a k-column $C = \begin{pmatrix} c_1 \\ c_2 \\ \cdot \\ \cdot \\ c_k \end{pmatrix}$ consisting of 0's and 1's among n participants in such a way that the vector can be retrieved only when all

n participants cooperate. We begin by making a set of group generators $X = \{x_1, \dots, x_m\}$ public. The scheme is as follows:

(1) The dealer distributes over a secure channel to each participant P_j a set of words R_j in the alphabet $X^{\pm 1} = \{x_1^{\pm 1}, \dots, x_m^{\pm 1}\}$ such that each group $G_j = \langle x_1, \dots, x_m \mid R_j \rangle$ has efficiently solvable word problem.

(2) The dealer splits the secret bit column C (the actual secret to be shared) into a sum $C = \sum_{j=1}^{n} C_j$ of n bit columns; these are secret shares to be distributed.

(3) The dealer then distributes words $w_{1j}, \dots, w_{kj}$ in the generators $x_1, \dots, x_m$ over an open channel to each participant P_j, $1 \le j \le n$. The words are chosen so that $w_{ij} \ne 1$ in G_j if $c_i = 0$ and $w_{ij} = 1$ in G_j if $c_i = 1$.

(4) Participant P_j then checks, for each i, whether the word $w_{ij} = 1$ in G_j or not. After that, each participant P_j can make a column of 0's and 1's, $C_j = \begin{pmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{kj} \end{pmatrix}$, by setting $c_{ij} = 1$ if $w_{ij} = 1$ in G_j and 0 otherwise.

(5) The participants then construct the secret by forming the column vector $C = \sum_{j=1}^{n} C_j$, where the sum of the entries is taken modulo 2.

In Step (5) of the above protocol, the participants can use secure computation of a sum as proposed in [2] if they do not want to reveal their individual column vectors, and therefore their individual secret shares, to each other. In order to implement the protocol to compute a secure sum, the participants should be able to communicate over secure channels with one another. These secure channels should be arranged in a circuit, say, $P_1 \to P_2 \to \dots \to P_n \to P_1$. Then the protocol to compute a secure sum is as follows:

(1) P_1 begins the process by choosing a random column vector N_1. He then sends to P_2 the sum $N_1 + C_1$.

(2) Each P_i, for $2 \le i \le n-1$, does the following. Upon receiving a column vector C from participant P_{i-1}, each participant P_i chooses a random column vector N_i and adds $N_i + C_i$ to C and sends the result to P_{i+1}.

(3) Participant P_n chooses a random column vector N_n and adds $N_n + C_n$ to the column he has received from P_{n-1} and sends the result to P_1. Now P_1 has the column vector $\sum_{i=1}^{n} (N_i + C_i)$.

(4) Participant P_1 subtracts N_1 from what he got from P_n; the result now is the sum $S = \sum_{1 \le i \le k} C_i + \sum_{2 \le i \le k} N_i$. (This step is needed for P_1 to preserve privacy of his N_1, and therefore of his C_1, since P_2 knows $N_1 + C_1$.) Then P_1 broadcasts S to other participants.

(5) The participants then pool together to recover the secret. They do this by each subtracting his random column vector N_i, $2 \le i \le n$, from S.

Thus, by using n secure channels between the participants, the participants are able to compute a secure sum in this secret sharing scheme. For more on the computation of a secure sum see [2].

3.1. **Efficiency.** We note that the dealer can efficiently build a word w in the normal closure of R_i as a product of arbitrary conjugates of elements of R_i, so that $w = 1$ in G_i. Furthermore, if G_i is a *small cancellation group* (see our Section 6), then it is also easy to build a word w such that $w \neq 1$ in G_i: it is sufficient to take care that w does not have more than half of any cyclic permutation of any element of R_i as a subword. See our Section 6 for more details. Finally, we note that in small cancellation groups (these are the platform groups that we propose in this paper), the word problem has a very efficient solution, namely, given a word w in the generators of a small cancellation group G, one can determine, in linear time in the length of w, whether or not $w = 1$ in G.

4. A (t, n)-threshold Scheme

Here we propose a scheme that combines Shamir's idea with our scheme in Section 3. As in Shamir's scheme, the secret is an element $x \in \mathbb{Z}_p$, and the dealer chooses a polynomial f of degree $t-1$ such that $f(0) = x$. In addition the dealer determines integers $y_i = f(i)$ (mod p) that are to be distributed to participants P_i, $1 \leq i \leq n$. A set of group generators $\{x_1, \ldots, x_m\}$ is made public. We assume here that all integers x and y_i can be written as k-bit columns. Then the scheme is as follows.

(1) The dealer distributes over a secure channel to each participant P_j a set of relators R_j such that each group $G_j = \langle x_1, \ldots, x_m | R_j \rangle$ has efficiently solvable word problem.
(2) The dealer then distributes over open channels k-columns
$b_j = \begin{pmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{kj} \end{pmatrix}$, $1 \leq j \leq n$, of words in $x_1, \ldots, x_m$ to each participant. The words b_{ij} are chosen so that, after replacing them by bits (as usual, "1" if $b_{ij} = 1$ in the group G_j and "0" otherwise), the resulting bit column represents the integer y_j.
(3) Participant P_j then checks, for each word b_{ij}, whether or not $b_{ij} = 1$ in his/her group G_j, thus obtaining a binary representation of the number y_j, and therefore recovering y_j.
(4) Each participant now has a point $f(i) = y_i$ of the polynomial. Using polynomial interpolation, any t participants can now recover the polynomial f, and hence the secret $x = f(0)$.

If $t \geq 3$, then the last step of this protocol can be arranged in such a way that participants do not have to reveal their individual shares y_i to each other if they do not want to. Indeed, from the Lagrange interpolation formula we see that

$$f(0) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t,\ j \neq i} \frac{-j}{i-j}.$$

Thus, $f(0)$ is a linear combination of private y_i with publicly known coefficients $c_i = \prod_{1 \le j \le t, j \ne i} \frac{-j}{i-j}$. If $t \ge 3$, then this linear combination can be computed without revealing y_i, the same way the sum of private numbers was computed in our Section 3.

In the special case $t = 2$, this yields an interesting problem. Note that in the original Shamir's scheme, pairs $(i, f(i))$ of coordinates are sent to participants over secure channels, so that the second coordinates are private, whereas the first coordinates are essentially public because they just correspond to participants' numbers in an ordering that could be publicly known. This, however, does not have to be the case, i.e., the first coordinates can be made private, too, so that the dealer sends private points $(x_i, f(x_i))$ to participants. Then, for $t = 2$, we have the following problem of independent interest:

Problem 1. *Given that two participants, P_1 and P_2, each has a point (x_i, y_i) in the plane, is it possible for them to exchange information in such a way that at the end, they both can recover an equation of the line connecting their two points, but neither of them can recover precise coordinates of the other participant's point?*

5. Why Use Groups?

One might ask a natural question at this point: "What is the advantage of using groups in this scheme? Why not use just sets of elements R_j as long-term secrets, and then distribute elements w_{ij} that either match some elements of R_j or not?" The disadvantage of this procedure is that it will eventually compromise the secrecy of R_j because matching elements will have to be repeated sooner or later. On the other hand, there are infinitely many different words that are equal to 1 in a given group G. For example, if $w = 1$ in G, then also $\prod_i h_i^{-1} w h_i = 1$ for any words h_i. Thus, the dealer can send as many words $w_{ij} = 1$ in G to the participants as he likes, without having to repeat any word or update the relators R_j.

The question that still remains is whether some information about relators R_j may be leaked, even though the words distributed over open channels will never match any words in R_j. This is an interesting question of group theory; we address it, to some extent, in our Section 7.

6. Platform Group

In order for our scheme to be practical, we need each participant to have a finite presentation of a group with efficiently solvable word problem. Here we suggest small cancellation groups as a platform for the protocol. For more information on small cancellation groups see e.g. [3].

Let $F(X)$ be the free group on generators $X = \{x_1, \ldots, x_n\}$. A word $w(x_1, \ldots, x_n) = x_{i_1}^{\epsilon_1} \cdots x_{i_n}^{\epsilon_n}$, where $\epsilon_i = \pm 1$ for $1 \le i \le n$, is called *cyclically reduced* if it is a reduced word and $x_{i_1}^{\epsilon_1} \ne x_{i_n}^{-\epsilon_n}$.

A set R containing cyclically reduced words is called *symmetrized* if it is closed under taking cyclic permutations and inverses. Given a set R of relators, a non-empty word $w \in F(X)$ is called a *piece* if there are two distinct relators $r_1, r_2 \in R$ such that w is an initial segment of both r_1 and r_2; that is, $r_1 = wv_1$ and $r_2 = wv_2$ for some $v_1, v_2 \in F(X)$ and there is no cancellation between w and v_1 or w and v_2.

In the definition below, $|w|$ denotes the lexicographic length of a word w.

Definition 1. *Let R be a symmetrized set of relators, and let $0 < \lambda < 1$. A group $G = \langle X; R\rangle$ with the set X of generators and the set R of relators is said to satisfy the small cancellation condition $C^{'}(\lambda)$ if for every $r \in R$ such that $r = wv$ and w is a piece, one has $|w| < \lambda|r|$. In this case, we say that G belongs to the class $C^{'}(\lambda)$.*

We propose groups that satisfy the small cancellation property because groups in the class $C^{'}(\frac{1}{6})$ have the word problem efficiently solvable by Dehn's algorithm. The algorithm is straightforward: given a word w, look for a subword of w which is a piece of a relator from R of length more than a half of the length of the whole relator. If no such piece exists, then $w \neq 1$ in G. If there is such a piece, say u, then $r = uv$ for some $r \in R$, where the length of v is smaller than the length of u. Replace the subword u by v^{-1} in w, and the length of the resulting word will become smaller than that of w. Thus, the algorithm must terminate in at most $|w|$ steps. This (original) Dehn's algorithm is therefore easily seen to have at most quadratic time complexity with respect to the length of w. We note that there is a slightly more elaborate version of Dehn's algorithm that has linear time complexity.

We also note that a generic finitely presented group is a small cancellation group (see e.g. [5]); this means, a randomly selected set of relators will define a small cancellation group with overwhelming probability. Therefore, to randomly select a small cancellation group, the dealer in our scheme can just take a few random words of length > 6 and check whether the corresponding symmetrized set satisfies the condition for $C'(\frac{1}{6})$. If not, then repeat.

7. Tietze transformations: elementary isomorphisms

This section is somewhat more technical than the previous ones. Our goal here is to show how to break long defining relators in a given group presentation into short pieces by using simple isomorphism-preserving transformations. This is useful because in a small cancellation presentation (see our Section 6) defining relators tend to be long and, moreover, a word that is equal to 1 in a presentation like that should contain a subword which is a piece of a defining relator of length more than a half of the length of the whole relator. Therefore, exposing sufficiently many words that are equal to 1 in a given presentation may leak information about defining relators. On the other hand, if all defining relators are short (of length 3, say), a word that is equal to 1 in such a presentation is indistinguishable from random.

Long time ago, Tietze introduced isomorphism-preserving elementary transformations that can be applied to groups presented by generators and defining relators (see e.g. [3]). They are of the following types.

- **(T1):** *Introducing a new generator*: Replace $\langle x_1, x_2, \ldots \mid r_1, r_2, \ldots\rangle$ by $\langle y, x_1, x_2, \ldots \mid ys^{-1}, r_1, r_2, \ldots\rangle$, where $s = s(x_1, x_2, \ldots)$ is an arbitrary word in the generators $x_1, x_2, \ldots$.
- **(T2):** *Canceling a generator* (this is the converse of (T1)): If we have a presentation of the form $\langle y, x_1, x_2, \ldots \mid q, r_1, r_2, \ldots\rangle$, where q is of the form ys^{-1}, and $s, r_1, r_2, \ldots$ are in the group generated by $x_1, x_2, \ldots$, replace this presentation by $\langle x_1, x_2, \ldots \mid r_1, r_2, \ldots\rangle$.

- **(T3):** *Applying an automorphism*: Apply an automorphism of the free group generated by $x_1, x_2, \dots$ to all the relators $r_1, r_2, \dots$.
- **(T4):** *Changing defining relators*: Replace the set $r_1, r_2, \dots$ of defining relators by another set $r'_1, r'_2, \dots$ with the same normal closure. That means, each of $r'_1, r'_2, \dots$ should belong to the normal subgroup generated by $r_1, r_2, \dots$, and vice versa.

Tietze has proved (see e.g. [3]) that two groups $\langle x_1, x_2, \dots \mid r_1, r_2, \dots \rangle$ and $\langle x_1, x_2, \dots \mid s_1, s_2, \dots \rangle$ are isomorphic if and only if one can get from one of the presentations to the other by a sequence of transformations (T1)–(T4).

For each Tietze transformation of the types (T1)–(T3), it is easy to obtain an explicit isomorphism (as a mapping on generators) and its inverse. For a Tietze transformation of the type (T4), the isomorphism is just the identity map. We would like here to make Tietze transformations of the type (T4) recursive, because *a priori* it is not clear how one can actually apply these transformations. Thus, we are going to use the following recursive version of (T4):

(T4′) In the set $r_1, r_2, \dots$, replace some r_i by one of the: r_i^{-1}, $r_i r_j$, $r_i r_j^{-1}$, $r_j r_i$, $r_j r_i^{-1}$, $x_k^{-1} r_i x_k$, $x_k r_i x_k^{-1}$, where $j \neq i$, and k is arbitrary.

Now we explain how the dealer can break down given defining relators into short pieces. More specifically, he can replace a given presentation by an isomorphic presentation where all defining relators have length at most 3. This is easily achieved by applying transformations (T1) and (T4′), as follows. Let Γ be a presentation $\langle x_1, \dots, x_k; r_1, ..., r_m \rangle$. We are going to obtain a different, isomorphic, presentation by using Tietze transformations of types (T1). Specifically, let, say, $r_1 = x_i x_j u$, $1 \leq i, j \leq k$. We introduce a new generator x_{k+1} and a new relator $r_{m+1} = x_{k+1}^{-1} x_i x_j$. The presentation $\langle x_1, \dots, x_k, x_{k+1}; r_1, \dots, r_m, r_{m+1} \rangle$ is obviously isomorphic to Γ. Now if we replace r_1 with $r'_1 = x_{k+1} u$, then the presentation $\langle x_1, \dots, x_k, x_{k+1}; r'_1, \dots, r_m, r_{m+1} \rangle$ will again be isomorphic to Γ, but now the length of one of the defining relators (r_1) has decreased by 1. Continuing in this manner, one can eventually obtain a presentation where all relators have length at most 3, at the expense of introducing more generators.

We conclude this section with a simple example, just to illustrate how Tietze transformations can be used to cut relators into pieces. In this example, we start with a presentation having two relators of length 5 in 3 generators, and end up with a presentation having 5 relators of length 3 in 6 generators. The symbol $\cong$ below means "is isomorphic to".

Example. $\langle x_1, x_2, x_3 \mid x_1^2 x_2^3,\ x_1 x_2^2 x_1^{-1} x_3 \rangle \cong \langle x_1, x_2, x_3, x_4 \mid x_4 = x_1^2,\ x_4 x_2^3,\ x_1 x_2^2 x_1^{-1} x_3 \rangle$
$\cong \langle x_1, x_2, x_3, x_4, x_5 \mid x_5 = x_1 x_2^2,\ x_4 = x_1^2,\ x_4 x_2^3,\ x_5 x_1^{-1} x_3 \rangle$
$\cong \langle x_1, x_2, x_3, x_4, x_5, x_6 \mid x_5 = x_1 x_2^2,\ x_4 = x_1^2,\ x_6 = x_4 x_2,\ x_6 x_2^2,\ x_5 x_1^{-1} x_3 \rangle$.

We note that this procedure of breaking relators into pieces of length 3 increases the total length of relators by at most the factor of 2.

8. Conclusions

We have proposed a two-stage (t, n)-threshold secret sharing scheme where long-term secrets are distributed to participants over secure channels, and then shares of the actual secret can be distributed over open channels.

Our scheme has the same distributed secret as Shamir's scheme does, but rather than sending shares of a secret over secure channels, we send the integers in disguise (as tuples of words in a public alphabet) over open channels, and then participants use group-theoretic methods to recover the integers. This scheme has the following useful advantages over Shamir's original scheme:

- The actual secret need not be distributed over secure channels and, furthermore, once the long-term private information is distributed to all participants (over secure channels), several different secrets can be distributed without updating the long-term private information.
- While recovering the secret, participants do not have to reveal their shares to each other if they do not want to.

References

[1] B. Fine, M. Habeeb, D. Kahrobaei, and G. Rosenberger. Survey and open problems in non-commutative cryptography. *JP Journal of Algebra, Number Theory and Applications*, 21:1–40, 2011.
[2] D. Grigoriev and V. Shpilrain. Unconditionally secure multiparty computation and secret sharing. *preprint.*
[3] R. Lyndon and P. Schupp. *Combinatorial Group Theory.* Classics in Mathematics. Springer, 2001.
[4] A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based cryptography.* Advanced Courses in Mathematics. CRM Barcelona. Birkhuser Verlag, Basel, 2008.
[5] A. Yu. Ol'shanskii. Almost every group is hyperbolic. *Internat. J. Algebra Comput.*, 2:1–17, 1992.
[6] D. Panagopoulos. A secret sharing scheme using groups. *Preprint, 2010. Available at http://arxiv.org/PScache/arxiv/pdf/1009/1009.0026v1.pdf.*
[7] A. Shamir. How to share a secret. *Comm. ACM*, 22:612–613, 1979.
[8] D. R. Stinson. *Cryptography: Theory and Practice.* Chapman and Hall, 2006.

CUNY Graduate Center, City University of New York
E-mail address: MHabeeb@GC.Cuny.edu

CUNY Graduate Center, City University of New York
E-mail address: DKahrobaei@GC.Cuny.edu

The City College of New York and CUNY Graduate Center
E-mail address: shpil@groups.sci.ccny.cuny.edu